VALIDATING BEHAVIORAL MODELS FOR REUSE[1]

Richard J. Malak Jr.
gtg269k@prism.gatech.edu

Christiaan J.J. Paredis*
chris.paredis@me.gatech.edu

Systems Realization Laboratory
G.W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332

*Corresponding author: tel: +1 404 894 5613; fax: +1 404 894 9342

Abstract

When using a model to predict the behavior of a physical system of interest, engineers must be confident that, under the conditions of interest, the model is an adequate representation of the system. The process of building this confidence is called model validation. It requires that engineers have knowledge about the system and conditions of interest, properties of the model and their own tolerance for uncertainty in the predictions. To reduce time and costs, engineers often reuse preexisting models that other engineers have developed. However, if the user lacks critical parts of this knowledge, model validation can be as time consuming and costly as developing a similar model from scratch. In this article, we describe a general process for performing model validation for reused behavioral models that overcomes this problem by relying on the formalization and exchange of knowledge. We identify the critical elements of this knowledge, discuss how to represent it and demonstrate the overall process on a simple engineering example.

Keywords: model validation, model reuse, model context.

[1] To appear in a forthcoming issue of Research in Engineering Design. The original publication is available at www.springerlink.com.

1 Introduction

In this investigation, we explore how engineering designers can have confidence in a mathematical model for the behavior of a system when the model was developed by someone else. The general process of establishing this confidence is known as *model validation* [1]. Although there exists much literature regarding model validation, most of this work focuses on situations in which it is possible to make empirical observations of the system being modeled. This typically is not the case in the context of engineering design, where designers seek to evaluate a number of hypothetical systems and usually will realize only one of these physically (and only after evaluations are needed). Another limitation of prior work on model validation is that most approaches described in the literature are incapable of accounting for the fragmentation of knowledge that can occur in cases of reuse, where the engineers who developed a model and the engineers who seek to reuse it each have different types of knowledge and all of this knowledge is critical to model validation. It is possible to overcome the limitations of prior work by reexamining the problem at the knowledge level and recasting model validation as a problem in knowledge exchange. We describe a general process for performing model validation in reuse scenarios that accounts for the actual knowledge requirements and constraints of model reuse in the context of engineering design.

The basic motivation for this investigation is the potential value of reusing behavioral models (also called engineering analysis models). By reusing a model, designers avoid the potentially expensive and time-consuming task of developing a new model. This fact is not lost on designers, and informal reuse strategies are part of common practice. Designers often adapt models published in reference books, copy computer code that implements part or all of their model (a practice known in the computer engineering community as "code scavenging"), and invoke software components they have written or purchased previously (e.g., calling library routines, interfacing to a remote service). However, these practices by-in-large are not part of a systematic reuse-focused modeling and simulation strategy, and they do not account for the special challenges of validating reused behavioral models.

Recent advances in the areas of information technology and engineering software make it easier for engineers to reuse behavioral models, but do little to support model validation. A key conclusion of the current investigation is that the problem is conceptual rather than technical. Moreover, existing information technology can support model validation for reused behavioral models in the context of engineering design if the validation problem is reformulated in the appropriate conceptual framework, which we develop in this article. The essential question is: *what knowledge must the user of a behavioral model have in order to perform model validation?* The main contribution is a novel process by which an engineering designer can perform model validation when seeking to reuse a behavioral model created by another engineer and lacking empirical data about the system being modeled.

Prior work on model validation for cases of model reuse focuses on having model developers constrain future reuse of a model by specifying the situations for which they have validated it. This can prevent invalid reuse, but it also prevents reuse in situations that actually are valid but were not anticipated or investigated by model developers. Model users determine how a model will be used and must deal directly with the consequences of model validity and therefore it is more appropriate for them to draw final conclusions about model validity. Another limitation of prior work is that it focuses on documentation as a means for knowledge exchange. Although documentation is useful and important, it is insufficient in light of current trends in information technology. A computer-interpretable representation of knowledge is desirable to support automation in the search, retrieval, instantiation and execution of behavioral models.

The organization of this paper is as follows. Section 2 is an overview of behavioral model reuse in the context of engineering design. Section 3 is a review of the model validation literature, including the basic objectives, foundational principles, key concepts and the limitations of current approaches in the context of model reuse for engineering design problems. Section 4 is a description of the proposed model validation process and its foundations in the knowledge requirements of model users. Section 5 is a demonstration of the process on an illustrative engineering example. Section 6 is a discussion of significance and limitations of the process and concepts described in this article.

2 Behavioral Model Reuse

2.1 Behavioral Modeling and Simulation

In this article, the term behavioral model refers to any mapping of what a system *is* to what it *does* under a particular set of circumstances. This is in contrast to models that represent other aspects of a design, such as its form (i.e., what it is) or function (i.e., intended behavior). One can use the terms *engineering analysis model* and *behavioral model* interchangeably, and many engineers use the unqualified term *model* with the same meaning.

Engineers use behavioral models to make predictions about a system without having to observe it physically. This is important when the system cannot be observed or observations are impractical. In engineering design, the system of interest typically is a design alternative that exists only in concept. By using behavioral models, engineers can evaluate a large number of design alternatives quickly and inexpensively, focusing their resources on prototyping only the most promising alternatives.

The term *simulation* refers to the process of obtaining predictions from a model, which sometimes is referred to as *solving* a model or performing *design analysis*. Some prefer to reserve the term simulation to refer strictly to solving time-dependent behavioral models, but no such distinction is made here. If a behavioral model is sufficiently simple in structure, an engineer may be able to solve it by hand—possibly arriving at a general closed-form result. In most cases, some level of computer support is helpful. Computer-based approaches range from implementing a solution algorithm or heuristic in a programming language to using a domain-specific tool that handles most solver-related details automatically.

The process of creating a new behavioral model varies from case to case and the terms *model development*, *modeling* and *model generation* all refer to this process. For some problems model development involves extensive software engineering. This is more common for highly specialized problems where commercial tools are inadequate, typically involving models of extreme size or mathematical complexity. Such problems require customized simulation code, and the model and the method for solving it typically are tightly intertwined.

For many engineering problems it is possible for one to use commercial tools to simplify model development. Rather than implementing a solver themselves, engineers can formulate a behavioral model in a way that it can be interpreted by a general-purpose solver, which are available for a wide range of modeling formalisms. This eliminates most software engineering issues, and simplifies model development. Some tools go further by generating the appropriate equations given a description of a system. For example, many finite element analysis packages work this way, generating and solving a set of equations given the general form of the constitutive equations, a description of artifact geometry, and boundary conditions.

2.2 Behavioral Model Reuse in Engineering Design

Motivation for Reuse

The motivation for model reuse is that of value: by reusing a preexisting behavioral model designers can save the time and expense of developing a new one, and the more times they reuse a model the smaller its per-use cost. If a model is reused only once or twice, the overhead associated with reuse (storage, search services, etc.) may undercut any gains. However, the potential exists for widespread behavioral model reuse in the context of engineering design problems. The systems that engineers design can vary greatly in their details, but the physical principles upon which the systems work—the laws of motion, thermodynamics, etc.—are common to all engineered systems. In principle, engineering designers can model the behavior of their design alternatives by reusing preexisting generic models that are based on physical principles. While it remains unclear whether engineers can realize such a strategy at a reasonable cost (i.e., so that there is a net value in model reuse), the potential for this exists and prompts further investigation.

Model Reuse in the Literature

Several authors discuss the topic of model reuse without providing an explicit definition for the term. This undoubtedly is a consequence of the everyday familiarity of the notion of reuse. In this article, *behavioral model reuse* refers to the use of a preexisting behavioral model for a simulation study for which its validity initially is unknown to its user. The term implies nothing about the intent of its developers or about its prior uses. Note that behavioral model reuse does not generally equate with the simulation of a behavioral model more than one time; one can consider multiple executions of a model during the course of a simulation study (e.g., trade study, sensitivity analysis) to constitute a single use of a model.

Some authors define model reuse as any use of an existing model to aid in the development of a new model (e.g., [2]). This parallels the accepted definition for software reuse (e.g., [3]) likely because software engineering plays a significant role in some model development processes. Such a definition is reasonable from a model development perspective, but has limitations from a model validation perspective because it neglects the ultimate use of a behavioral model: to yield predictions about a system for use in decision making. Although the development of new models from preexisting ones is an important topic, it is outside the scope of this article—the focus here is on validation.

Behavioral Model Reuse in Engineering Practice

Behavioral model reuse already is common for engineering problems. There are several current examples of behavioral model reuse as well as research that promises to deliver increased levels of reuse:

- Several engineering analysis packages allow engineers to develop models from predefined component models and in many cases engineers can specialize the predefined models through their parameters. For example, many physics-based modeling tools (e.g., Dymola, AMESim) ship with libraries of generic models for different phenomena (e.g., electrical, mechanical, hydraulic). Engineers can model their system by instantiating a generic model with appropriate parameter settings.
- The element types in most commercial finite-element tools are generic models that one specializes with parameter values appropriate for the system being modeled (typically, FEA software handles the instantiation and parameter specialization automatically). Engineers typically may not think of them as such, but each element in a mesh is one instantiation of a generic element model (e.g., 3D elastic beam or 3D thermal solid) with parameter values appropriate for the given geometry, material and boundary conditions.
- Extensive research on the integration of preexisting models exists within the discrete event systems and U.S. Defense Modeling and Simulation Office (DMSO) communities. The High-Level Architecture (HLA) is a standardized framework for integrating multiple distributed simulation applications [4, 5]. It allows one to reuse a model by incorporating it into another, higher-level model.
- In the engineering domain, commercially available tools exist that help engineers integrate preexisting behavioral models (e.g., ModelCenter, iSIGHT). These tools allow engineers to link independent, distributed behavioral models in a black-box fashion and, unlike the HLA, allow one to integrate legacy models that were not developed with reuse in mind.
- Advances in repository technology promises to provide engineers with easier access to a wider array of behavioral models [6, 7]. These systems allow engineers to search a distributed database for models having specified properties.

The open question about behavioral model reuse is not whether engineers will reuse models, but how valuable they can make the practice. Technologies exist that lower the cost of reusing a model or allow one to reuse models in previously impractical situations, both of which add to the value of reuse. However, improper reuse of behavioral models can undermine these gains—the consequences of a bad decision made with an invalid model easily can outweigh the benefits of having reused the model.

Current Limitations

The primary limitation of current technology for behavioral model reuse is one of knowledge representation. This is evident when one considers the elements of a reuse strategy. Researchers in the software engineering community identify three generic steps in a software reuse process: selection, adaptation and integration [8]. These also hold true in the context of behavioral modeling. The selection step is highly dependent on the information one has available, which makes the abstract description of a reusable artifact a key facet of any reuse strategy. This results in four dimensions along which one can classify a reuse strategy [3, 9]. The dimensions—adapted to the case of behavioral modeling—are:

- Abstraction: How one describes the capabilities and limitations of a behavioral model.
- Selection: How one locates, compares and selects behavioral models.
- Specialization: How (or whether) one is permitted to modify a behavioral model.
- Integration: How one combines selected behavioral models into a simulation framework (and possibly with other behavioral models).

Each of these dimensions is observable in current model reuse technology. For example, the HLA and some commercial engineering tools cited above provide integration capabilities. Model libraries allow specialization through tunable parameters. Repositories allow users to search distributed databases of models using complex queries.

A major problem with behavioral model reuse is that advances in model abstraction are not keeping pace with those in other areas. Although information and knowledge management technology enables engineers to formalize their knowledge, the technology provides no guidance on what knowledge to represent. It is this area that most significantly impacts model validation. Consequently, it has become easier for engineers to reuse behavioral models but remains difficult for them to perform model validation in those cases.

3 Model Validation

3.1 Objectives and Motivations

The principal motivation for performing model validation is to establish a degree of credibility for a model. This is crucial when one wishes to use a model in support of decision making. Essentially, bad models can lead to bad information which, in turn, can lead to bad decisions.

Model validation is about adequacy, not correctness. All models are abstractions of reality and, as such, are incorrect (in that they differ from observation) under some circumstances. One seeks to establish whether a model is "correct enough" for one's needs, and it is meaningless to say that one model is "more valid" than another. More precisely, we define model validation as the process of determining whether using a model in a particular simulation study yields computational results with sufficiently small uncertainty over the set of study scenarios. Numerous variations of this definition exist in the literature, most of which are similar in substance (see [10, 11] for broader terminology surveys).

The uncertainty of a model reflects how much the model limits what one can know about the system being modeled given the simulation results. The relationship between uncertainty and model validity is indirect, and depends on how one will use the simulation results. Often, engineers can validly use a model for one decision but not another because the decisions have differing requirements. For example, engineers may deem a model valid for making a decision in conceptual design where knowledge about the precise behavior of the system is unnecessary but deem that same model invalid for a detail design decision.

Seldom is it desirable for engineers to use the most precise model available, as reducing uncertainty typically costs more in terms of development and computational expense. Ideally, engineers seek to use the least expensive model that is adequate for the given problem. The threshold defining adequacy is subject to one's value judgment and can be subject to information economic tradeoffs such as those explored in [12].

3.2 Fundamental Basis

Model validation is a subjective, scientific process that is open to refutation but not positive justification. Moreover, one can prove that a model is invalid for a particular use but cannot prove the converse in a definitive sense; validity is something one concludes subjectively after weighing all the available evidence. This basic nature of model validation has several practical consequences, including that it places a premium on expert judgment, domain knowledge and trust among collaborators. However, it does not mean that model validation must be unscientific or arbitrary—a proper model validation process is rooted in observation (either directly of the system being modeled or indirectly through derivation of a model from other models (e.g., physical laws) that themselves have been scrutinized relative to empirical data) and judgments are non-arbitrary in the sense that they must not contradict the observations or any other accepted knowledge about the problem.

This perspective on model validation is based on the philosophies of science and knowledge. Naylor and Finger [13] generally are credited with being the first to interpret the model validation problem in a philosophical context, and several others have extended their ideas (e.g., [14-16]).

That no amount of data can "prove" a model to be valid is a consequence of a classic problem in philosophy, originally described by Charles Hume and known as the *problem of induction* [17]. The problem is essentially one of having to generalize beyond observational data (i.e., what is known to be true). For example, to prove a continuous relationship between two variables requires an infinite amount of data because the relationship is defined at an infinite number of locations. Clearly this is impossible and any "proof" based on finite data requires one to make additional assumptions (e.g., about the smoothness of the relationship). The modern resolution of the problem of induction is based on Karl Popper's notion of *falsification*, which states that one can demonstrate the falsity of a scientific claim using empirical evidence but failure to falsify such a claim can lead only to one's accepting the claim as true and does not constitute proof of its veracity [18]. Furthermore, one's acceptance of a claim as true is provisional, pending future attempts at refutation.

That model validation is necessarily subjective follows from the falsification perspective, according to which one must subject a claim (in this case, a model) to extensive attempts at refutation and accept it as valid only after such attempts fail to refute it. Subjectivity arises primarily in terms of which tests to perform and for how long to perform tests before accepting the model as valid. No general rule exists, so engineers must use their judgment as domain experts to decide whether additional testing is warranted.

3.3 Basic Conceptual Framework

Extensive research into model validation exists within the discrete-event simulation and DMSO communities. Although work from these communities is not always directly applicable to problems in the engineering design domain, it is useful for establishing the basic objectives, concepts and knowledge requirements for model validation. To serve as a conceptual guide to understanding and performing model validation, researchers have developed a framework that relates the basic entities and processes involved in model development and validation. Figure 1 is an illustration of the framework, which is based on an abstract conceptualization of the modeling and simulation process for discrete-event simulations. It derives from the work of Sargent [19], whose work draws from earlier work by the Society for Computer Simulation Technical Committee on Model Credibility [1]. Other, more elaborate, frameworks exist in the literature, but the abstract depiction in Figure 1 includes the fundamental aspects of model validation while being clear and understandable [20]. This process is generic enough to be consistent with behavioral modeling and simulation for engineering problems, as we explain below. The key characteristic of this framework from the perspective of the current investigation is that it highlights the information important to model validation.

There are several processes through which one can build confidence in the validity of a model, each of which provides a different path through which information can flow. *Conceptual model validation* (also called theoretical validation or model qualification) is the process of substantiating that the theories and assumptions underlying a conceptual model—a verbal, logical or mathematical description of a system—are correct and that the representation of the system is reasonable for the intended simulation study. This requires knowledge about the conceptual model, the system being represented and the requirements of the simulation study (experiment scenarios and required uncertainty). *Model verification* is the process of assuring that a computerized model is a sufficiently

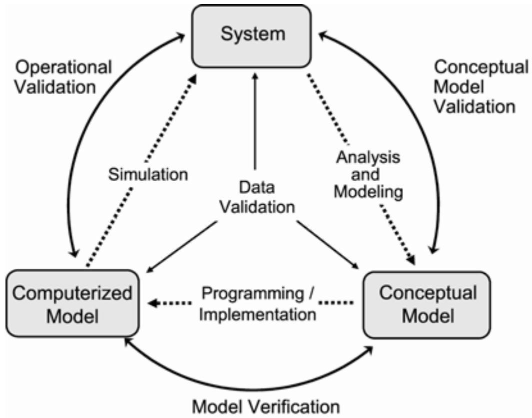

Figure 1: A conceptual framework for model validation based on an abstract modeling and simulation process [19].

accurate representation of the corresponding conceptual model. *Operational validation* is the process of determining whether the computerized model is sufficiently accurate for user needs for the scenarios of the simulation study. This typically involves comparisons of the input-output behavior of the model with observational data and requires knowledge about the computerized model, the system and the simulation study. *Data validation* is the process of ensuring that all numerical data used to support the other processes is accurate and consistent.

Due to the state of the art in engineering analysis tools, engineers may more commonly think of a single model development or model generation process rather than the analysis/modeling and programming/implementation steps from Figure 1. Many tools allow engineers to generate a computerized model automatically given a conceptual description of the problem (e.g., mesh generation for finite element analysis) and others permit engineers to specify computerized models in a form nearly identical to the corresponding conceptual model (e.g., equation-based modeling languages). Model verification is relatively straightforward in such cases, but remains a key challenge when engineers implement their own computerized models in programming languages such as C or Fortran.

There are two paths through which one can relate a computerized model to the system it represents: operational validation or conceptual model validation followed by model verification. In general, one can base conclusions about model validity on information from both paths. However, practical considerations often dictate what is feasible. For engineering design problems, operational validation often is infeasible because the system of interest is a design alternative that has not been constructed and the reason for performing a simulation study is to avoid doing so. In such situations, engineers can perform model validation relative to a suitably defined referent. A *referent* is a codified body of information that describes the characteristics and behavior of the system to be represented [21]. Thus, engineers can use existing theory and historical data from similar systems to guide validation of a model for a new system. A simple engineering example of this is the use of data from test specimens of a material. By combining this data with an understanding of how the specimen differs from the actual part in their system (e.g., surface finish or loading conditions), engineers can reason about the validity of a behavioral model for that part without having to build and test it.

3.4 Limitations in Cases of Model Reuse

From a procedural standpoint, the challenge of model validation is to bring together the various sources of knowledge that engineers require to draw conclusions about model validity. Model reuse makes validation difficult by disrupting the typical flow of knowledge. Most approaches to model validation based on the framework of Figure 1 have limitations that are problematic in cases of model reuse:

- They assume all knowledge for model validation is available as needed.
- They result in conclusions about validity for an overly-narrow scope of model uses.
- They often are highly reliant on text-based documentation.

Engineers cannot proceed with model validation without knowledge about the system being modeled, the limitations of a model, the experiment scenarios of a simulation study and the tolerable level of uncertainty in simulation results. Yet, in cases of model reuse, it is possible that neither the engineers who develop a model nor those reusing it have all this knowledge. Knowledge about a model and its limitations originates with its developers, whereas knowledge about the system being modeled, the simulation study and uncertainty requirements originates with model users. What is more, model developers and model users can operate in independent processes with no overlap in time or personnel. Thus, model validation approaches based on the framework of Figure 1—which presumes a single process with all knowledge present—are incomplete in cases of model reuse. To be effective in cases of reuse, *a model validation approach must account for the segmentation of knowledge between model developers and model users*.

Although it is possible for model developers to circumvent their lack of knowledge about future model uses, to do so requires them to make assumptions that can limit future reuse. Under such an approach, developers validate a model against their own intentions for model use, which permits valid reuse only for applications of the model that reflect developer intentions [22]. This approach finds application in practice (e.g., the DMSO Modeling and Simulation Resource Repository), but such developer-driven approaches are inefficient from a reuse perspective. The range of possible valid uses of an engineering behavioral model—all reasonable simulation studies and uncertainty requirements—is too broad for developers to consider in this way, which leads to artificial constraints on model reuse and diminishes the value of reuse. To be effective while not inhibiting reuse, *a model validation approach for engineering design must permit model users to draw conclusions about validity*.

Despite being better-suited for drawing conclusions about model validity, model users still can lack knowledge important to validation and must have model developers communicate this knowledge to them. Many approaches to model validation favor text-based documentation as a means for communicating knowledge. This has the benefit of being effective with less formality because humans with domain expertise can interpret what others have written. However, it presents practical challenges in an increasingly automated and distributed engineering environment. In cases where automation can be beneficial—such as when selecting a model with particular properties from a large repository—strict reliance on text-based documentation can be a hindrance. Language and semantic barriers also present problems as distributed engineering technology extends collaboration across traditional boundaries. To be effective in cases of reuse, *a model validation approach should supplement text-based documentation with a formal, computer-interpretable representation of the required knowledge* [7, 23].

To summarize, prior approaches to model validation have limitations in the context of reused behavioral models, but these limitations point to a remedy involving the formalization of knowledge relevant to validation and exchange of this knowledge directed from model developers to model users. The remaining question regards precisely what knowledge model developers must formalize and communicate to the users of their models, and the answer to this follows from the definition of model validation and the typical relationship between model developers and model users. Model users understand the details of the system being modeled and their tolerance for uncertainty in the simulation results, but can lack knowledge about the behavioral model. Specifically, they need to know its uncertainty for the situations in which they will use it. This knowledge will enable them to determine whether the model is sufficient for their needs. The following section is a description of a procedural framework for formalizing and using this knowledge.

4 Validation for Behavioral Model Reuse

4.1 Formalizing Validation Knowledge

Engineers seeking to reuse a behavioral model can perform model validation if the developers of the model communicate to them formalized descriptions of their knowledge about model *uncertainty* in a well-defined set of circumstances, or *context*. We call this formal description of knowledge about a model a *validity description*. With a proper validity description, model users can draw sound conclusions about the validity of a model for specific simulation studies. The process for accomplishing this is described in Section 4.4, however, first we introduce the foundational concepts involved in this validation process.

The need to formalize knowledge about model uncertainty follows directly from the definition for model validation cited in Section 3.1—users must determine whether a model yields predictions with sufficiently small uncertainty for their needs. The need to formalize knowledge about context derives from the fact that uncertainty is not an intrinsic property of a model, but a relative property that depends on the circumstances under consideration. Consider that a laminar flow model is a more accurate representation (i.e., contributes less uncertainty to predictions) of fluid systems with low fluid speeds than those with high speeds. A proper validity description for such a model includes both an characterization of its uncertainty and a definition of the context in which the uncertainty is characterized (e.g., specific fluid speeds).

Given the notion of a validity description, one can decompose model validation into three core activities.

> **Validity Characterization**: The process of developing a validity description.
>
> **Compatibility Assessment**: The process of determining whether the context of a behavioral model is consistent with that of the simulation study of interest.
>
> **Adequacy Assessment**: The process of determining whether the uncertainty of a behavioral model is acceptably small for simulation study objectives.

Model developers are responsible for performing validity characterization. They do this concurrently with model development, where they make decisions that impact model uncertainty over various contexts. Although developers can be involved in the other activities, this involvement is unnecessary assuming they communicate to model users an appropriate validity description.

Model users are responsible for both of the assessment activities. Compatibility assessment is a necessary precondition for adequacy assessment because the uncertainty reported in a validity description is meaningful only within the corresponding context. If that context is incompatible with the simulation study (in the sense described in the following section), then the actual model uncertainty in the situation of interest is unknown and users cannot draw conclusions about validity. Otherwise, users can proceed with adequacy assessment and compare their tolerance for uncertainty with the model's uncertainty guarantees to determine whether it is valid for their study.

Validity descriptions also can apply to the information involved in a simulation study (e.g., parameters, initial and boundary conditions). Inputs to a model may be uncertain estimates of physical quantities and their use can contribute to the uncertainty of simulation results. By associating validity descriptions with problem quantities, engineers can reason about the predictions that result from a simulation study. This relates to model validation because model users might require different levels of model uncertainty depending on the uncertainty of the input data they have available. Also, as we discuss in the next section, the contexts of all models and information used in a simulation study must be compatible to ensure that the resulting predictions are meaningful under the circumstances of interest to model users.

4.2 The Context of Validation Knowledge

4.2.1 What it Means for Validation Knowledge to have Context

Few statements or rules are universally true. While often left implicit, qualifiers exist that indicate the limitations of information and knowledge. When people communicate, they either presume a common understanding of a domain of discourse or state their assumptions explicitly. Communication is ambiguous when the assumption of common understanding is incorrect. The term *context* refers to the limited domain over which a statement of knowledge applies as dictated by the assumptions, idealizations and implications associated with it.

The importance of context is recognized in several communities. In the artificial intelligence community, several researchers discuss the importance of formalizing context for knowledge-based systems and propose means to do so (e.g., [24-26]; see [27] and [28] for surveys). Although the representations differ from what is proposed here, the basic motivations and objectives are similar. Context also is recognized in the modeling and simulation community as an important consideration for model reuse [29] and model validation [30], but little work exists on representation formalisms.

In the case of model validation for reuse, context relates claims about model uncertainty to a set of physical circumstances under which the claims hold. Context allows model users to reason about uncertainty in a sound manner and is necessary because model uncertainty depends on the physical situation of interest. Consider for example a model to predict the position of a body assuming its velocity is constant. One can express this as

$$\mathbf{x}(t) = \mathbf{v}_0 t + \mathbf{x}_0 , \qquad (1)$$

where t is time, $\mathbf{x}_0$ is the initial position vector, $\mathbf{v}_0$ is the (constant) velocity vector and $\mathbf{x}(t)$ is the predicted position. Any statement about the uncertainty of this model is relative to a set of physical circumstances: developers could quote model uncertainty relative to a system with negligible acceleration (i.e., near constant velocity) or relative to one with significant acceleration, the former uncertainty being much smaller than the latter. Potential model users need to know the context in which developers determine model uncertainty in order to determine whether they can reason about validity soundly. For example, engineers wishing to predict the behavior of a system undergoing large accelerations cannot reasonably conclude that the model in Equation (1) is adequate for their needs based on an uncertainty measured relative to a system with near-zero acceleration—this would be an underestimate of the model uncertainty relative to the system of interest to the engineers.

One also can employ the concept of context to describe the extent of a user's simulation study. For a given study, users seek to make predictions about the system of interest under a particular set of physical circumstances. This set of circumstances defines the context of the simulation study and must be compatible (in a sense defined later in this article) with that of a model if users are able to draw conclusions about validity based on the associated model uncertainty.

4.2.2 Representing the Context of Validation Knowledge

Explicit and formal representation of context is essential to ensure proper transfer of knowledge between engineers. Although engineers often understand context implicitly in their work and few would mistakenly use Equation (1) to model a body under large accelerations, models can have many subtle context restrictions that engineers might recognize only after significant analysis—if at all. This is underscored in a recent study in which investigators asked a small group of engineers of various skill levels to identify all the idealizations incorporated into a model for a body falling through a fluid [30].

Despite the relative simplicity of the model (a single ordinary differential equation), none of those surveyed identified more than 75% of the assumptions.

There has been much work on context formalization within the artificial intelligence community, but this work has limitations from the standpoint of behavioral model validation. The general approach they take is to represent assumptions about the world as statements in a logic. Falkenhainer and Forbus apply this approach to behavioral model components [31]. The basis for formalizing context using assumptions comes from the mechanics of mathematical modeling. Model developers make simplifications such as assuming a derivative is exactly zero or that a system is completely closed. Thus, one way to formalize these assumptions is to define logical statements that indicate them. For example, one might express the context of Equation (1) using the proposition `ConstantVelocity = true`.

Although logical formulations of assumptions provide a convenient means for reasoning about context at a high level, they require all participants to adhere to consistent semantics. This is problematic when representing the modeling assumptions for behavioral model reuse. Precise mathematical assumptions are convenient idealizations of a physical situation and seldom are satisfied exactly. Models that incorporate such assumptions are useful as long as the assumptions correspond "closely enough" to reality. The challenge for communicating context is capturing the semantics of an assumption. Although the mathematical model of Equation (1) includes the assumption that velocity is constant, it is reasonable in practice to use the model as long as accelerations are small. This is important because in reality accelerations rarely are exactly zero and the model therefore would never be applicable. Model developers understand what it means for an assumption to be met "closely enough" and must communicate this to model users in an unambiguous fashion.

Model developers can communicate the semantics of their assumptions using a set-based approach for representing context. Conceptually, a context defines a set of "world states" within which one has some assurance of correctness or accuracy and beyond which no such assurances exist. In principle, a context specifies allowable values of every variable in the "world." However, according to the principle of near-decomposability, only a handful of relevant variables affects a system in practice [32]; all others have so little impact on a model's predictions that they can be assumed practically unbounded.

In most cases, one can represent context using interval bounds on the values of physical quantities, which results in a hypercube in the variable space. Consider for example a context for Equation (1). The model assumes constant velocity, which is equivalent to zero acceleration. Thus, one can form a context as bounds on the acceleration magnitude, $\left\|\mathbf{a}(t)\right\|_2$ where $\mathbf{a}(t) = d\mathbf{v}(t)/dt$, $\mathbf{v}(t)$ is body velocity as a function of time and $\left\|\cdot\right\|_2$ is the Euclidean norm. By bounding the acceleration magnitude, one has a context of $\left\|\mathbf{a}(t)\right\|_2 \le a_{\max}$. This formulation of context conveys to users what developers mean by being "close enough" to constant velocity and therefore solves the ambiguity problem. Users of this model can be assured that any statements about its uncertainty are trustworthy for accelerations up to a_{max}, and they can use the model at that uncertainty level as long as the system being modeled does not exceed this limit.

Note that in the preceding example the context involves a quantity that does not appear explicitly in the model. This is because context is an expression of limitations on the *physical* circumstances and therefore potentially involves quantities not present in the model. Typically, this is a consequence of using a model that involves a simplifying assumption relating to that quantity, as illustrated in the example problem of Section 5.

There may be circumstances in which a variable-bounds approach to context representation is insufficient, particularly when one is under pressure to define a context as narrowly as possible. In such cases, one can use more general set-based representations to define the allowable values of a quantity. In general one faces a tradeoff between representational expressiveness and the computational challenges of reasoning with that representation and, therefore, should use the simplest representation that is reasonable.

4.2.3 Reasoning about the Context of Validation Knowledge

What it means for Contexts to be Compatible

As part of the model validation process, engineers must reason about the context of a model relative to that of the simulation study in which they will use it. Typically, the objective of a simulation study in engineering design is to generate predictions about behavioral attributes of a system for use in making decisions about the system. Each behavioral attribute can have its own context that is defined by the study objectives. For example, an engineer might require one prediction about structural stress under steady-state conditions and another about the probability of failure under specific dynamic conditions.

The predictions that result from using a model may have a different context than what engineers require for that behavioral attribute. Compatibility assessment is the process of determining whether or not this is the case. Failure to consider the context of a model can lead to unjustified engineering decisions. As an example consider the design of a supersonic aircraft. It would be risky for engineers to make design decisions based on subsonic performance predictions because these predictions are outside of the context of the supersonic behavior. By reasoning about context formally, engineers can ensure that design decisions are justified based on the given information.

In general, *one can conclude that a model is valid if and only if the context of the model is the same or broader than that of the behavioral predictions one wishes to make with it*. Otherwise, there will be some circumstances to which the model and its predictions do not apply. This is illustrated in Figure 2. While in Figure 2(a) the model yields

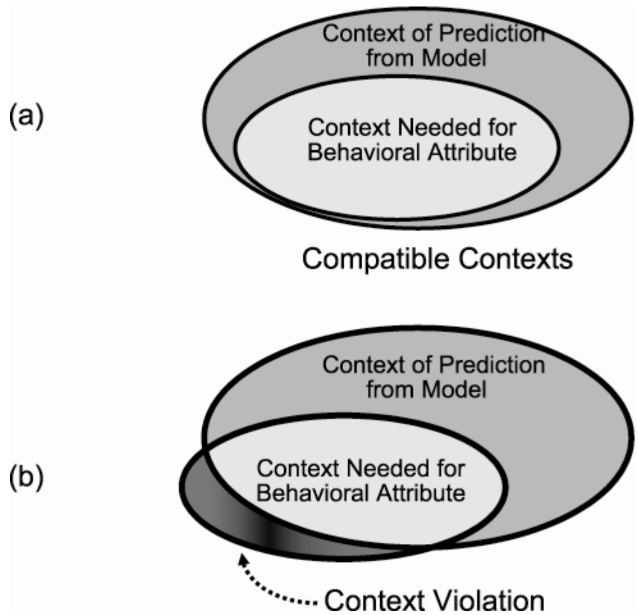

Figure 2: Contexts in a design decision: (a) the behavioral attribute context is subsumed by that of the corresponding prediction; (b) the behavioral attribute context is not subsumed by the context of the corresponding prediction.

predictions that span the desired context—i.e., the model is context-compatible with the study—one cannot validly use the model in the situation depicted in Figure 2(b). To use a model in this situation is akin to extrapolating beyond known data—a practice that typically is inadvisable.

In cases such as Figure 2(b), it may be possible to expand the context at the expense of greater model uncertainty. However, this is a model development task and model users may lack the resources to determine the uncertainty of the larger context.

The definition of a simulation experiment identifies a particular model, initial conditions and parameter values to be used. For a design problem, parameters specialize a behavioral model to a particular design alternative (i.e., they specify physical dimensions or other quantities that remain constant throughout the simulation) and inputs represent external stimuli and boundary conditions. The uncertainty of each element of a simulation experiment is associated with a particular context and the context of a prediction made by the simulation is the intersection of these contexts. Figure 3 is an illustration of the relationship of a prediction context to the contexts in a simulation experiment. Intuitively, a prediction cannot "know" more than the elements from which it was formed. For example, one cannot generally make valid predictions about turbulent

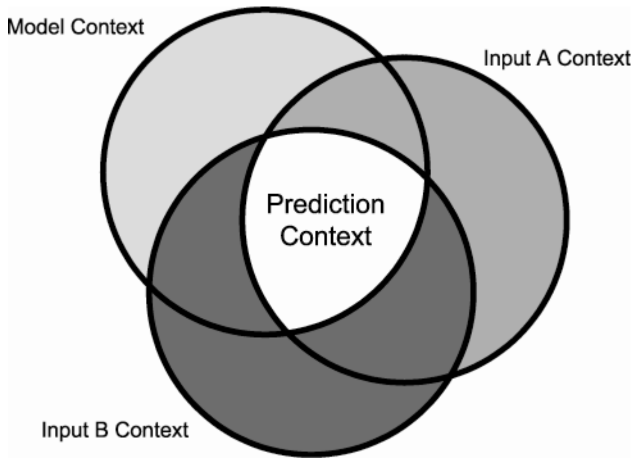

Figure 3: A depiction of the relationship between the context of a prediction (unshaded region) and those of the model and inputs (A and B) from which it is formed.

fluid flow based solely on a laminar flow model. Mathematically, one can state the general relationship as

$$C_P = C_M \cap \left(\bigcap_{j=1}^{N} C_j \right), \tag{2}$$

where C_P is the context of the prediction, C_M is the model context and C_j is the context of the j^{th} input or parameter to the model. This means that the context of a prediction is never more general than the least general context from which it is formed.

Assessing Context Compatibility

One can assess the compatibility of a model on two levels. In both cases, the assessment operations involve set comparisons (intersection and subsumption operations), the implementations of which depend on the chosen representation for context. For interval-based representations of context (as used in the example of Section 5), such operations are simple functions of the upper and lower interval bounds.

The more basic level of context compatibility is to ask whether a model is compatible with a given simulation experiment. To answer this question, one compares the context of a model to those of the parameters and inputs of an experiment. A model is compatible with the other elements of the experiment if the intersection of these contexts—i.e., the context of the resulting prediction, C_P of Equation (2)—is not the empty set. Assessing compatibility in this way can be useful in exploratory situations (e.g., engineers seeking to obtain a qualitative understanding of a design space or system behavior, perhaps to help them better formalize design decisions or other simulation studies) or as an intermediate step in assessing compatibility as defined below.

Commonly in design, one performs a simulation study to predict a specific behavioral attribute for use in a decision. In this case, one performs compatibility assessment for a model relative to whether a resulting prediction is context-compatible with the behavioral attribute. Given inputs and parameters for a model and a desired behavioral attribute, the use of a behavioral model is valid if the prediction yielded by the simulation experiment is context-compatible with the behavioral attribute. This combines the concepts illustrated in Figure 2 and Figure 3. Combining the notion of context compatibility with the relationship in Equation (2), one can state a rule for context compatibility for simulation studies. A model is context compatible with a simulation study if and only if

$$C_{BA} \subseteq \left(C_M \cap \left(\bigcap_{j=1}^{N} C_j \right) \right), \tag{3}$$

where C_{BA} is the required behavioral attribute context, C_j is the context of the j^{th} input or parameter (defined as part of the simulation study) and C_M is the model context.

In the case of interval-based representations, the context of each variable is expressed using an upper and lower bound on its permitted values and any unspecified variables are presumed unbounded. Each of the contextualized entities (the model, inputs, parameters and attributes to be predicted) can have their context defined by one or more bounded variables and, typically, multiple entities will have bounds for the same variables. Thus, one can evaluate Equation (3) by comparing the bounds.

Suppose a problem involves a model with its context defined by bounds on one variable, expressed as $\left[\underline{\theta}_M, \overline{\theta}_M\right]$, and the model has N total inputs and parameters, each with a context defined by bounds on the same variable and expressed as $\left[\underline{\theta}_j, \overline{\theta}_j\right]$, $j = 1 \ldots N$. To evaluate the intersections on the right-hand side of Equation (3), one must find the greatest lower bound and least upper bound for the variable. Let $\underline{\theta'}$ and $\overline{\theta'}$ be the lower and upper bounds, respectively, of the context on variable θ after taking the intersections. Thus,

$$\left[\underline{\theta'}, \overline{\theta'}\right] = \left[\max\left\{\underline{\theta}_M, \underline{\theta}_1, \ldots, \underline{\theta}_N\right\}, \min\left\{\overline{\theta}_M, \overline{\theta}_1, \ldots, \overline{\theta}_N\right\}\right].$$

To complete the evaluation of Equation (3), one must then compare the upper and lower bounds after intersection with the desired context of the attribute to be predicted. If $\left[\underline{\theta}_{BA}, \overline{\theta}_{BA}\right]$ are the desired bounds on variable θ for the behavioral attribute, then context compatibility holds if and only if

$$\left(\underline{\theta}_{BA} \geq \underline{\theta'}\right) \wedge \left(\overline{\theta}_{BA} \leq \overline{\theta'}\right). \tag{4}$$

That is, the interval defining the context of the behavioral attribute must be contained within the interval defined by the intersection of the model and input contexts.

Extending this procedure to multiple variables is straightforward. All computations proceed independently for each variable, and context compatibility holds true if and only if Equation (4) holds for *all* variables.

4.3 Uncertainty as Essential Validation Knowledge

4.3.1 What it Means for a Behavioral Model to be Uncertain

Whereas context relates to the breadth of conclusions one can draw from evidence, uncertainty relates to the strength of those conclusions, with less uncertainty enabling stronger conclusions. Behavioral models are uncertain in the sense that no matter what simulation study users conduct with a model, they will have a limited state of knowledge about the behavior of the real system. The role of a statement about uncertainty in a validity description is to convey to potential users the limitations of a model so that they can use it only when sufficient for their needs.

From a decision-theoretic standpoint, uncertainty is what prevents a decision maker from identifying an action that will result in the most preferred outcome [33]. In a model reuse scenario, the decision maker is a model user. To identify the best action, a model user would ideally know the exact consequences of all possible actions. This is not generally possible due to complications arising from, among other things, inherent randomness, lack of information or knowledge, and limited computational resources. Although behavioral models are not the only source of uncertainty in a decision, they are critical contributors and it is therefore important from a decision-making perspective to have a good characterization of their uncertainty.

It is common to conceive of uncertainty as having two different components: *variability* due to randomness and *imprecision* due to lack of information or knowledge [34-36]. The motivation for this distinction is that, conceptually, a decision maker can reduce imprecision by gathering more information or knowledge but is unable to affect variability. Both uncertainty components arise in modeling and simulation problems and therefore are important for model validation. Imprecision arises due to the assumptions and simplifications engineers make during model development, while variability commonly results from measurement processes and sometimes is inherent in the system being modeled.

Some authors prefer a more granular decomposition of uncertainty, particularly imprecision, decomposing imprecision into components based upon specific causes such as limited sample data, simplifications made during modeling and human error [36-39]. Such decompositions may be useful as a conceptual guide to model developers when constructing a validity description, but there is no need to report the disaggregate to users.

4.3.2 Representing Uncertainty for Use as Validation Knowledge

When formalizing model uncertainty in a validity description, model developers must take care to include the uncertainty due to all contributing factors. Failure to do so can lead model users to assess a model as adequate for their needs when it is not. In turn, this can lead users to draw conclusions that are unsupported at the true uncertainty level.

Several representations for uncertainty are possible, and which representation is most appropriate depends upon the uncertainties involved. In cases where the variability component dominates, a probabilistic representation of uncertainty is most appropriate. This is true for example when model developers are able to compare the model against empirical data containing random error. In cases where imprecision dominates, the use of interval bounds is more appropriate. For example, this would be the case when developers characterize the uncertainty of a deterministic model relative to a deterministic referent, such as when deriving a specialized model from a more general one. The example of Section 5 includes such a case.

There may be times when neither uncertainty component dominates—i.e., both imprecision and variability are significant and of the same order of magnitude. Pure probabilistic representations are inappropriate in such cases because they neglect the imprecision component of uncertainty. This implies less uncertainty than actually is present and could lead to invalid model use. Pure interval representations would not lead to invalid uses of a model, but they are overly conservative and can preclude uses that actually are valid. The best option from the standpoint of validation for model reuse is to use a hybrid representation that incorporates both imprecision and variability. Walley describes a theory of imprecise probabilities, which extends traditional probability theory to allow for intervals of probabilities [40]. Although suitable in principle, there are significant computational challenges associated with general imprecise probabilities. Ferson and Donald have developed a formalism, called probability bounds analysis (PBA), based on imprecise probabilities which includes additional restrictions that limit its expressiveness but improve its computational efficiency [41]. Recent work has shown PBA to be useful for engineering design problems [12, 42], making it an appropriate representation for uncertainty for use in validity descriptions.

4.3.3 Reasoning about Uncertainty for Model Validation Problems

Users perform adequacy assessment by comparing their tolerance for uncertainty to the uncertainty stated in the validity description for a model. A model is adequate if its uncertainty is "less than" the maximum tolerable uncertainty. One's interpretation of "less than" can depend on the uncertainty representation one adopts. When one represents uncertainty using classical probability theory, it is appropriate to make decisions about adequacy by applying the hypothesis tests described in most general texts on probability and statistics. For example, consider a model defined as $y = f(\mathbf{x}) + \varepsilon$,where $\mathbf{x}$ is an $n \times 1$ input vector (the inputs and parameters), y is the resulting prediction and ε is an error term modeled as a random variable with zero mean and variance σ^2. Assuming $f(\cdot)$ and $\mathbf{x}$ are deterministic then y is a random variable also with variance σ^2, and one can perform statistical tests using estimates of σ^2 to determine whether the prediction error is sufficiently small. The tests are more complicated in cases where the input vector, $\mathbf{x}$, also is nondeterministic. Users must propagate the input uncertainty through the model in order to obtain good estimates of the prediction uncertainty, and in some cases this will require executing the model (e.g., Monte Carlo simulation).

When including interval-based representations for imprecision, one must apply interval-based reasoning methods. In this case, one could have a model defined as $y = f(\mathbf{x}) + \delta$, where δ is an interval-valued variable with bounds $\left[\underline{\delta}, \overline{\delta}\right]$ (typically such that $\underline{\delta} < 0 < \overline{\delta}$). Assuming $f(\cdot)$ and $\mathbf{x}$ are precise (i.e., not interval-valued), then the prediction uncertainty also is $\left[\underline{\delta}, \overline{\delta}\right]$ and the model will be adequate if $\max\left\{\left|\underline{\delta}\right|, \left|\overline{\delta}\right|\right\} \leq \delta_{\text{tol}}$, where δ_{tol} is the level of uncertainty one can tolerate. As with a probabilistic representation, evaluation of prediction uncertainty is more complicated in cases involving uncertainty in the model inputs, $\mathbf{x}$, and can require one to execute the model.

For most models and predictions, their uncertainty depends upon the context in question. For instance, a linear deflection model for a beam may be very accurate (i.e., low uncertainty) when the displacement is less than some upper bound, but inaccurate otherwise (i.e., high uncertainty). In general, uncertainty never decreases—and likely increases—as the context expands. This results in an important tradeoff for model developers: too narrow a context can yield a model with low uncertainty but that is seldom useful, while too broad a context can result in a model too uncertain to be useful.

4.4 General Process Flow

In Figure 4, we present a flow chart for the overall model validation process for behavioral model reuse introduced in this article. The validation steps (light gray) are depicted as part of a model development and use processes. The process flow includes no direct interactions between model developers and model users, though there may be cases in which they interact more directly. Figure 4 represents the most challenging situation from a model validation standpoint—because they are not involved in its development, users know nothing about the validity-related properties of a model beyond what developers report in a validity description.

At the end of model developer activities, their model and its corresponding validity description is published to a repository or other suitable search and retrieval system. Although developers might treat model development and validity characterization as sequential steps or as involving some degree of iteration, they are closely related and viewing them as concurrent processes is reasonable at this level of abstraction (step D2). Note that step D2 is analogous to the model development and validation process of Figure 1, but with validity characterization replacing model validation.

Model users begin by defining their simulation study (step U1) and proceed to locate, validate and use a model. Users may repeat steps U2-U7 multiple times until they find a model that is valid for their study. In steps U3 and U4, users eliminate models that are invalid based on the initial conditions and general characteristics of their simulation study. However, it often is impossible to conclude definitively that a model is valid until after conducting one or more simulation runs. For example, the system's state in a

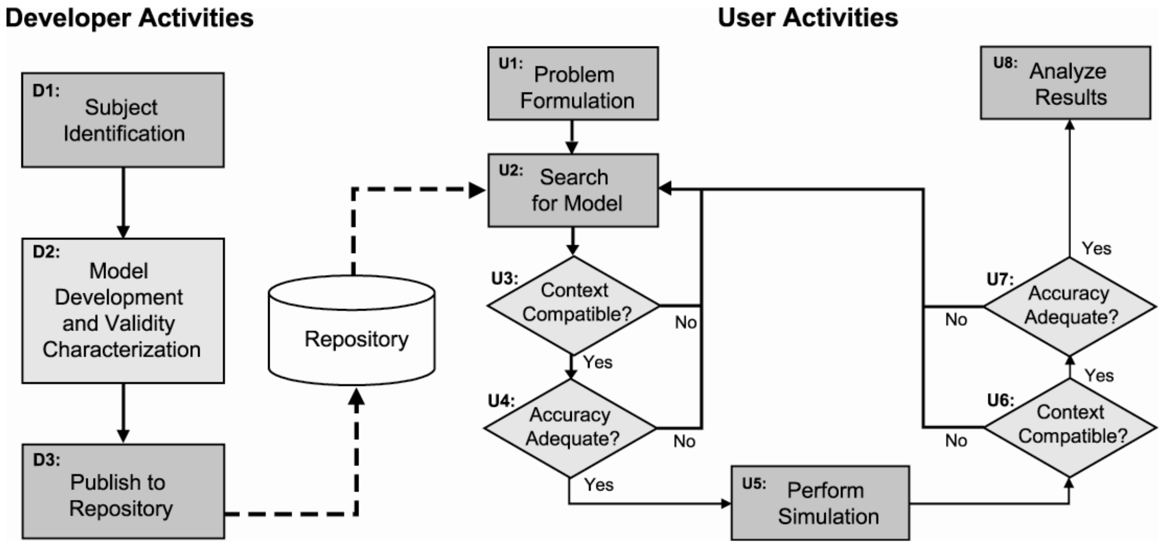

Figure 4: Process flow of validation for behavioral model reuse. Light-gray boxes are model validation activities; dark-gray boxes are other activities in the model development and reuse processes.

dynamics simulation might begin within the context of a model but stray beyond it as the simulation run progresses. Users should catch most invalid models in step U3 or U4, but steps U6 and U7 are necessary for confirmation.

5 Applying the Framework

The following is an example of how engineers can perform validation for behavioral model reuse within the framework described in this paper. The example involves model validation activities performed both by model developers (validity characterization) and model users (compatibility and adequacy assessment).

5.1 Demonstration Scenario

In the example scenario, engineers wish to determine the extension of a structural steel beam held statically in axial tension. Figure 5 is an illustration of the situation to be evaluated. A force, F, is applied to a beam of initial length, L_0 and initial cross-sectional area, A_0. Due to the force, the beam deforms to have length, L and the behavioral attribute of interest to the engineers is the change in beam length, ΔL. The engineers have knowledge about the beam (e.g., initial dimensions, material properties, etc.) and its operating conditions that can impact beam behavior (e.g., applied load, ambient temperature, etc.).

According to study objectives, the engineers need to predict beam extension to within ± 0.5 millimeters. They have designed the beam to be made of structural steel. The beam is 0.55 meters long with a cross-sectional area of 140 square millimeters. They know that their manufacturing process will yield dimensions up to $\pm 5\%$ of what is specified. The supplier of the structural steel reports to them that the material has a

Young's modulus of 200 ± 10 GPa and a coefficient of thermal expansion of $(12 \pm 0.5) \times 10^{-6}$ per Kelvin. The engineers also have determined details about the operating environment of the beam for the study. The ambient temperature may fluctuate but remains between 295 and 335 Kelvin. The static load applied to the beam is between 200 and 250 kN.

In order to evaluate the change in beam length, the engineers seek to reuse a model created by other engineers. This model is of a beam held in axial tension, but does not necessarily apply to the situation in question. The model developers perform validity

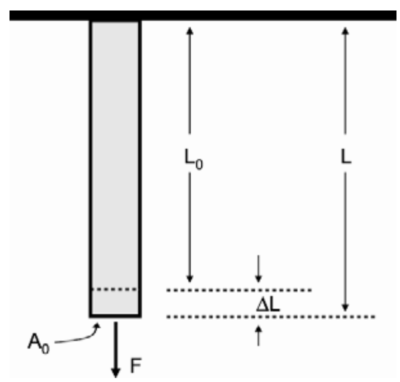

Figure 5: Illustration of a beam held in axial tension (deformation exaggerated).

characterization while they develop the model. Using the resulting validity description, the model users can perform compatibility and adequacy assessment to see whether the model applies to their situation.

To perform validity characterization, it is necessary to identify a context and to determine the uncertainty in that context. In practice, model developers likely would do this without specific knowledge of how the model would be used. For this example, a context is chosen that is appropriate for the system being modeled under some simulation scenarios that are likely, but differ from the simulation problem scenarios. This reflects the fact that model developers would lack this specific knowledge in practice. The uncertainty is determined by comparing the model with one of superior accuracy (i.e., a referent). Mathematically, the referent is assumed to be perfectly accurate and comparisons between the two models within the model context yields an absolute uncertainty for the new one. Although this approach involves strong assumptions from a practical perspective, it is adequate for illustrating the framework.

5.2 Characterizing the Validity of a Model

5.2.1 Model Development and Validity Characterization

Beam Extension Model

One possible model for the scenario of Figure 5 is based on Hooke's law, which relates stresses and strains in a material. Hooke's law typically is stated as

$$\sigma = E\varepsilon ,$$

where σ is the stress in a material, E is a material property known as the Young's modulus and ε is the strain in the material. Stress is defined as the force per unit area and strain is defined as the change in length per initial length. In relation to the terms in Figure 5, $\sigma = F/A_0$ and $\varepsilon = (L - L_0)/L_0$. By substituting these relationships and performing a few algebraic manipulations, one can arrive at the following model for extension in the beam:

$$L - L_0 = L_0 \frac{F}{EA_0} . \tag{5}$$

Validity Characterization

To be useful to engineers other than its developers, this model requires a validity description. To characterize the model, engineers require a baseline for comparison. For the present example, a model with less restrictive assumptions serves as a referent. In general, engineers can use empirical data and expert opinion in addition to higher-fidelity models.

The model of Equation (5) incorporates many assumptions. The impact of two particular assumptions are examined in detail here:

1. Thermal expansion is negligible.
2. The engineering definition of strain is an adequate approximation of the true strain.

The presence of these assumptions leads to uncertainty in the model outputs relative to actual system behavior. Engineers must convey the impact of these assumptions in a validity description.

The first assumption refers to the behavior that most materials exhibit as a result of temperature change. A typical model for this behavior has strain as a linear function of temperature change, $\varepsilon_T = \alpha\left(T_f - T_i\right)$, where ε_T is the thermal strain, T_f is the final temperature, T_i is the initial temperature and α is known as the coefficient of thermal expansion and depends on the material. Although more complex and precise relationships for this effect exist, this relationship is assumed perfectly accurate in this example. Thus, the total strain in a material is expressed as

$$\varepsilon_{\text{total}} = \varepsilon_{\sigma} + \varepsilon_{T} = \frac{\sigma}{E} + \alpha\left(T_f - T_i\right). \tag{6}$$

The model of Equation (5) involves only the strain due to loading conditions, ε_{σ}.

The second assumption refers to the distinction between "true" and "engineering" models for strain. The engineering definition used to formulate Equation (5) uses the initial length for strain, which is a good approximation for small deformations and is convenient because most of the information required to compute with them is based on the initial conditions of the system. The "true" definition (sometimes called the logarithmic definition) is more accurate over a wider range of circumstances and is sound under the superposition of strains. True strain is defined as $\tilde{\varepsilon} = \ln\left(L/L_0\right)$, where L is the final beam length [43].

Using the definition for true strain along with Equation (6), one can arrive at a model for extension in an axially loaded beam:

$$L - L_0 = L_0\left(e^{\left(\frac{F}{A_0 E} + \alpha\left(T_f - T_i\right)\right)} - 1\right).$$

We use this model as a referent for characterizing the simpler model of Equation (5). Using the absolute difference between the referent model and the Equation (5) as a measure of modeling error, one has

$$\delta_{\text{beam}} = \left| L_0\left(e^{\left(\frac{F}{A_0 E} + \alpha\left(T_f - T_i\right)\right)} - 1 - \frac{F}{A_0 E}\right)\right|, \tag{7}$$

where δ_{beam} is the modeling error due to the assumptions.

A validity description consists of an estimate of the uncertainty over a specified context. A context must include restrictions on all variables that influence the uncertainty. Failure to bound crucial variables can lead the uncertainty to be unbounded over the stated context. For Equation (5) the context must include all the variables appearing in Equation (7). The chosen variable bounds are given in Table 1. These values are representative of a long, slender rod made of structural steel.

The corresponding uncertainty is an upper bound of the magnitude of Equation (7) over this context. As reported in the validity description of Table 1, the uncertainty is less than a quarter of a millimeter for the specified context. This represents less than 10% of the maximum ΔL prediction possible within the context.

5.2.2 Remarks

The validity description of Table 1 is reached without knowledge of any particular model use. This is important from a model reuse perspective. Note that knowledge of a particular model use differs from general knowledge of how such a model might be used.

The latter is necessary for model developers to identify a relevant context and construct a model with a sufficiently small uncertainty so that it is likely to be useful to other engineers.

The validity description given in Table 1 is not unique. Model developers can specify any context they see fit based upon their domain experience. In turn, a different context can lead to a different uncertainty. Furthermore, developers can use other representations for context and uncertainty. If likelihood data is available, developers might choose a probabilistic uncertainty representation or one that can convey uncertainty due to both variability and non-random causes.

One interesting feature of the validity description is that it involves variables that are not part of the model in Equation (5) (e.g., temperature change, final beam length). This is necessary because they impact uncertainty. Their inclusion in the context is an indication that they influence system behavior but have been excluded due to simplifying assumptions. This is precisely the knowledge about models that users can lack, which is why it must appear in a validity description.

Table 1: Validity description for example problem model.

Model	$L - L_0 = L_0 \dfrac{F}{A_0 E}$	
Validity Description	Context	$T_f, T_i \in [290, 350]$ K $\alpha \in [11, 13]\ ^{10^{-6}}\!/_{\mathrm{K}}$ $E \in [180, 220]$ GPa $L_0, L \in [0.5, 1.0]$ m $A_0 \in [100, 150]$ mm^2 $F \in [100, 1000]$ kN
	Uncertainty	$\lvert \delta_{\text{beam}} \rvert \leq 0.24$ mm

5.3 Assessing the Compatibility and Adequacy of a Model

5.3.1 Compatibility Assessment

To perform compatibility assessment, model users compare the context encountered in their simulation study to that of the candidate model. In general, this can require performing a simulation because it may be impossible to determine precise conditions otherwise. This is reflected in the flow diagram of Figure 4. The current example involves a static situation so only a single comparison step is necessary.

The contexts of the study and model are stated in Table 2. For each quantity, the study context falls inside of the model context (i.e., Equation (4) holds for each variable). Thus, the model is context-compatible with the study. One would conclude that the context is not compatible if any one of the individual contexts are incompatible (e.g., if the model context for applied force was $[20,50]\,\text{kN}$, then the model would be incompatible with the study despite all other variables being compatible).

5.3.2 Adequacy Assessment

Since the model is context-compatible with the study, model users can proceed to evaluate its adequacy with respect to their tolerance for uncertainty. As is the case with compatibility assessment, the users can determine adequacy without executing the model. The users require predictions of beam extension to within ± 0.5 millimeters. They can determine model accuracy from its validity description, which is given in Table 1. Within the stated context, the model has an uncertainty of $\pm 0.24\text{ mm}$. The users can conclude that the model is adequate for their needs and therefore valid for their study.

Table 2: Comparison of model context of study context for compatibility assessment.

Quantity	Model Context [min, max]	Study Context [min, max]	Model Compatible?
Minimum Temperature	$[290,350]\,\text{K}$	$[295,335]\,\text{K}$	Yes
Maximum Temperature	$[290,350]\,\text{K}$	$[295,335]\,\text{K}$	Yes
Coefficient of Thermal Expansion	$[11,13]\ {}^{10^{-6}}\!/_{\text{K}}$	$[11.5,12.5]\ {}^{10^{-6}}\!/_{\text{K}}$	Yes
Young's Modulus	$[180,220]\,\text{GPa}$	$[190,210]\,\text{GPa}$	Yes
Initial Beam Length	$[0.5,1.0]\,\text{m}$	$[0.5225,0.5775]\,\text{m}$	Yes
Final Beam Length	$[0.5,1.0]\,\text{m}$	$[0.5225,0.5775]\,\text{m}$	Yes
Initial Cross-sectional Area	$[100,150]\,\text{mm}^2$	$[133,147]\,\text{mm}^2$	Yes
Applied Force	$[100,1000]\,\text{kN}$	$[200,250]\,\text{kN}$	Yes
Final Conclusion: Is model context-compatible with study?			Yes

5.3.3 Remarks

The assessment steps are carried out without knowledge of the inner workings of the model. Both steps are possible using only the knowledge formalized in the validity description of Table 1. This is crucial from a reuse perspective, as model developers may be unavailable to assist users during validation and users may be unable to understand or examine the model implementation.

Based upon the example given here, compatibility and adequacy assessment may seem straightforward compared to validity characterization. However, the simplicity of the assessment steps is contingent upon a proper formalization of the simulation study requirements. It is possible that model users will need to revise the definition of their simulation study once they become aware of which variables are included in the context definition of models they are considering using in their study.

Another simplifying factor in this example is that it is possible to draw conclusions about validity without evaluating the model. In terms of the general process depicted Figure 4, Steps U6 and U7 are unnecessary. This is possible only because the example involves an algebraic model. For more sohphisticated formalisms, one may be unable to determine whether the trajectory of a simulation will remain within the context of a model without performing the simulation. .

6 Discussion

The framework described in this article allows engineers to overcome the special challenges associated with performing validation for behavioral models in reuse scenarios. The three validation activities—validity characterization, compatibility assessment and adequacy assessment—serve to clarify and organize the roles of model developers and model users, ensuring that the appropriate validation knowledge is passed from model developers to the users of their models. The example of Section 5 is a demonstration of the framework and establishes the sufficiency of validity descriptions as a knowledge transfer mechanism—the user of a behavioral model can validate its use when his or her only knowledge about the model is its validity description. This is particularly important in a distributed design setting, where engineers can have remote access to behavioral models developed by others (e.g., via an internet-based repository) but may be unable to question the developers of a model about its characteristics. The validation knowledge we identify in this article is not always a consideration in remote collaboration tools, but it is necessary to enable efficient and low-risk model reuse. One prospective outcome of this thread of research would be the inclusion of validity descriptions in behavioral model repositories as a necessary part of the description of a model.

The proposed model validation framework is general and extends to different modeling formalisms and knowledge representations, even though our demonstration involves only algebraic models and an interval-based knowledge representation. This is evident in the fact that the framework is derived from the definition of model validation and the special requirements associated with behavioral model reuse, rather than the characteristics of any particular modeling formalism or knowledge representation. However, open questions remain regarding the relative value of the proposed framework when implemented using different formalisms and representations. For example, the interval-based representation of context used in this article allows one to draw inferences inexpensively (e.g., assessing context compatibility using a small number of bounds comparisons), but one is unable to express complex relationships in the variable space. Using more expressive representations may allow one to bound uncertainty more tightly, but it is unclear whether the resultant gains (e.g., wider reuse of the model) will offset any increases in the cost of formalizing and computing with the knowledge. A methodology for evaluating such tradeoffs would be valuable and it may be possible to develop one based on value of information theory [44], but this remains a topic for future investigation.

Even for a particular modeling formalism or knowledge representation, open questions remain regarding appropriate methods and methodologies for developing validity descriptions. Engineers must be able to identify which variables should be bounded in a context and how broad these bounds should be. The first problem is challenging because it deals with "unknown unknowns"—i.e., engineers must answer the question "is the uncertainty of my model significantly affected by variables I have not yet identified?" Once all such variables are identified, engineers face the task of bounding them broadly enough so that the validity description applies to many use situations but narrowly enough so that the corresponding uncertainty is reasonably small. Principles based on engineering or value of information considerations may help guide such decisions, but it seems likely that practical experience will be the most helpful.

Another open research question relates to how validity descriptions scale with the size and type of model. For some models the number of quantities bounded in a context description may be quite large, as suggested by a preliminary study [30]. Computing with large context descriptions can be expensive and engineers may be forced to make tradeoffs or to abstract frequently-encountered idealizations into a hierarchical context definition. For example, most engineering applications operate on the Earth and at speeds and sizes far from the threshold of relativistic or quantum effects. As such, the precise semantics of these conditions are not at issue and it may be adequate to abstract such conditions into logical conditions (e.g., `OnEarth = true`) and concentrate on precise formalization of only the factors that are likely to cause validity concerns (flow speed in fluids models, deflection magnitude for statics problems, etc.).

Although the proposed framework deals with the formalization and sound use of knowledge, it is important not to confuse this with an attempt to *prove* model validity. The objective of knowledge formalization within the framework is to serve as a basis for communication among engineers and traceability in a model validation process. Validity descriptions are an embodiment of the judgment of model developers (e.g., the size of a context, which uncertainty representation is best) and therefore are subjective. Model users also bring subjectivity into the process via their definition of a simulation study and their approach to performing the assessment steps (e.g., section of comparison methods, interpretation of "larger" for uncertainty). Thus, the framework does not constitute an effort to prove validity and is consistent with the philosophical basis of model validation discussed in Section 3.2.

The inherent subjectivity of model validation does raise some questions, particularly in a distributed design setting. In order for one engineer to reason with the subjective conclusions of another, there must exist a basis of trust between the engineers. Such trust does not come easily when the engineers are distributed globally, possibly working for different organizations and perhaps not even aware of one another's identities. For the proposed framework to be successful, model users must be able to trust the validity descriptions furnished by model developers. Although the framework does not address this need directly, the validation literature identifies two possible approaches for doing so that could be incorporated into the framework: accreditation and certification. According to the ISO definitions[2], accreditation is a process by which an authority recognizes formally that a person or organization meets some established competency level at a specific set of tasks and Certification is a complementary process in which one recognizes the conformance of a product, process or service to specific standards [46]. Accreditation helps to identify companies and individuals that meet minimum standards on some task, such as modeling in a particular domain. Certification increases a user's confidence that a particular result—a validity description, for instance—is as specified. Alternately, certification can apply to the methods used to develop particular results. The development of specific accreditation and certification procedures for the framework identified in this article is an issue for future development. However, it should be a somewhat direct extension of existing procedures to the specific activities associated with the framework.

7 Summary

This article introduces a new procedural framework for performing model validation for behavioral models in reuse scenarios. Existing methods are inappropriate for reuse scenarios because they do not address the gaps in knowledge that can arise between collaborating engineers. Although existing validation approaches are useful in some reuse cases, they undermine the benefits of reuse in the most general cases of engineering design problems. By relying on formal characterizations of model properties as a knowledge transfer mechanism, the framework avoids the knowledge-gap problem and allows engineers to validate behavioral models in a way that preserves the value of model reuse. Although several open questions exist relating to this work the framework remains a necessary and important step towards general behavioral model reuse.

Acknowledgement

The authors thank Jason Aughenbaugh, Jay Ling, Steven Rekuc, Morgan Bruns for their contributions to this work. This work is supported by the G.W.W. School of Mechanical Engineering at the Georgia Institute of Technology and NASA Ames Research Center under cooperative agreement NNA04CK40A.

References

[1] Schlesinger S, Crosbie RE, Innis GS, Lalwani CS, Loch J, Sylvester RJ, Wright RO, Kheir N and Bartos D (1979) Terminology for Model Credibility. *Simulation*. 32: 103-104.

[2] Robinson S, Nance RE, Paul RJ, Pidd M and Taylor SJE (2004) Simulation Model Reuse: Definitions, Benefits, and Obstacles. *Simulation Modeling Practice and Theory*. 12: 479-494.

[3] Krueger CW (1992) Software Reuse. *ACM Computing Surveys*. 24: 131-183.

[2] Readers should note that the terminology is not strictly uniform. The U.S. Department of Defense publishes a definition of accreditation that corresponds to the ISO definition for certification [45].

[4] HLA Working Group (2000) IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. Standard IEEE STD 1516-2000. IEEE, New York, NY.
[5] Morse KL, Lightner M, Little R, Liutz B and Scrudder R (2006) Enabling Simulation Interoperability. IEEE Computer. 39: 115-117.
[6] Szykman S, Sriram R, Bochenek C and Racz J (1998) The NIST Design Repository Project. In *Advances in Soft Computing - Engineering Design and Manufacturing*. Springer-Verlag, London.
[7] Mocko G, Malak Jr. RJ, Paredis CJJ and Peak R (2004) A Knowledge Repository for Behavioral Models in Engineering Design, in *Proceedings of the ASME Computers and Information in Engineering Conference*, Salt Lake City, UT.Paper No. CIE2004-57746.
[8] Basili VR, Briand LC and Melo W (1996) How Reuse Influences Productivity in Object-Oriented Systems. *Communications of the ACM*. 39: 104-116.
[9] Biggerstaff TJ and C. R (1989) Reusability Framework, Assessment, and Directions. In Biggerstaff TJ and Perlis AJ (eds) *Software Reusability: Volume 1: Concepts and Models*. Frontier Series. ACM Press, New York.
[10] Malak Jr. RJ (2005) A Framework for Validating Reusable Behavioral Models in Engineering Design. Masters thesis, Georgia Institute of Technology, Atlanta, GA.
[11] Oberkampf WL and Trucano TG (2002) Verification and Validation in Computational Fluid Dynamics. *Progress in Aerospace Sciences*. 38: 209-272.
[12] Ling J, Aughenbaugh JM and Paredis CJJ (2006) Managing the Collection of Information under Uncertainty using Information Economics. *Journal of Mechanical Design*. 128: 980-990.
[13] Naylor TH and Finger JM (1967) Verification of Computer Simulation Models. *Management Science*. 14: B92-B101.
[14] Barlas Y and Carpenter S (1990) Philosophical Roots of Model Validation: Two Paradigms. *System Dynamics Review*. 6: 148-166.
[15] Herskovitz PJ (1991) A Theoretical Framework for Simulation Validation: Popper's Falsificationism. *International Journal of Modeling and Simulation*. 11: 56-58.
[16] Kleindorfer GB and Ganeshan R (1993) The Philosophy of Science and Validation in Simulation, in *Proceedings of the Winter Simulation Conference*, Evans GW, Mollaghasemi M, Russell EC and Biles WE (eds) Institute of Electrical and Electronic Engineers, Piscataway, NJ. 50-57.
[17] Hume D (1965) *A Treatise of Human Nature* (1739-40 orig), Clarendon Press, Oxford.
[18] Popper KR (1972) *The Logic of Scientific Discovery* (1934 orig) 6th ed, Hutchinson, London.
[19] Sargent RG (1985) An Expository on Verification and Validation of Simulation Models, in *Proceedings of the 1985 Winter Simulation Conference*, San Francisco, CAGantz D, Blais G and Solomon S (eds) Institute of Electrical and Electronics Engineers, Piscataway, NJ. 15-22.
[20] Banks J, Gerstein D and Searles SP (1987) Modeling Processes, Validation and Verification of Complex Systems: A Survey, in *Proceedings of the Conference on*

Methodology and Validation, Orlando, FLBalci O (ed) Society for Computer Simulation. 13-18.

[21] Pace D (2004) The Referent Study Final Report. JWR-04-010. U.S. Defense Modeling and Simulation Office, Alexandria, VA.

[22] Balci O and Ormsby WF (2000) Well-defined Intended Uses: An Explicit Requirement for Accreditation of Modeling and Simulation Applications, in *Proceedings of the 2000 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ.

[23] Grosse IR, Milton-Benoit JM and Wileden JC (2005) Ontologies for Supporting Engineering Analysis Models. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*. 19: 1-18.

[24] Guha RV and Lenat DB (1992) Language, Representation and Contexts. *Journal of Information Processing*. 15: 340-349.

[25] Akman V and Surav M (1997) The Use of Situation Theory in Context Modeling. *Computational Intelligence*. 13: 427-438.

[26] McCarthy J (1993) Notes on Formalizing Context, in *Proceedings of the IJCAI-93: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, Chambery, France, 28 Aug.-3 Sept.Morgan Kaufmann Publishers. 555-560.

[27] Guha RV and McCarthy J (2003) Varieties of Contexts, in *Proceedings of the Modeling and Using Context, 4th International and Interdisciplinary Conference, CONTEXT 2003*, Stanford, CA, 23-25 JuneBlackburn P, Ghidini C, Turner RM and Giunchiglia F (eds) Springer. 164-177.

[28] Akman V and Surav M (1996) Steps Toward Formalizing Context. *AI Magazine*. 17: 55-72.

[29] Yilmaz L (2004) On the Need for Contextualized Introspective Models to Improve Reuse and Composibility of Defense Simulations. *Journal of Defense Modeling and Simulation*. 1: 141-151.

[30] Spiegel M, Reynolds Jr. PF and Brogan DC (2005) A Case Study of Model Context for Simulation Composibility and Reusability, in *Proceedings of the 2005 Winter Simulation Conference*, Kuhl ME, Steiger NM, Armstrong FB and Joines JA (eds) Institute of Electrical and Electronic Engineers.

[31] Falkenhainer B and Forbus KD (1991) Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence*. 51: 95-143.

[32] Simon HA (1996) *The Sciences of the Artificial* 3rd ed, MIT Press, Cambridge, MA.

[33] Clemen RT (1996) *Making Hard Decisions: An Introduction to Decision Analysis* 2nd ed, Duxbury Press, Pacific Grove, CA.

[34] Ferson S and Ginzburg L (1996) Different Methods are Needed to Propagate Ignorance and Variability. *Reliability Engineering and System Safety*. 54.

[35] Parry GW (1996) The Characterization of Uncertainty in Probabilistic Risk Assessment of Complex Systems. *Reliability Engineering and System Safety*. 54: 119-126.

[36] Oberkampf WL, DeLand SM, Rutherford BM, Diegert KV and Alvin KF (2000) Estimation of Total Uncertainty in Modeling and Simulation. SAND2000-0824. Sandia National Laboratories, Albuquerque, NM.

[37] United States Nulclear Regulatory Commission (1983) PRA Procedures Guide. Washington, DC.

[38] Der Kiureghian A (1989) Measures of Structural Safety under Imperfect States of Knowledge. *Journal of Structural Engineering*. 115: 1119-1139.

[39] Nikolaidis E (2005) Types of Uncertainty in Design Decision Making. In Nikolaidis E, Ghiocel DM and Singhal S (eds) *Engineering Design Reliability Handbook*. CRC Press, New York.

[40] Walley J (1991) *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, New York.

[41] Ferson S and Donald S (1998) Probability Bounds Analysis, in *Proceedings of the International Conference on Probabilistic Safety Assessment and Management (PSAM4)*, New York, NYSpringer-Verlag.

[42] Aughenbaugh JM and Paredis CJJ (2006) The Value of Using Imprecise Probabilities in Engineering Design. *Journal of Mechanical Design*. 128: 969-979.

[43] Shigley JE and Mischke CR (2001) *Mechanical Engineering Design* 6 ed, McGraw-Hill, New York, NY.

[44] Lawrence DB (1999) *The Economic Value of Information* Springer.

[45] United States Department of Defense (2003) DoD Modeling and Simulation (M&S) Verification, Validation and Accreditation (VV&A). DoD Instruction Number 5000.61, GPO, Washington, D.C., 24 Feb 2005 <http://www.dtic.mil/whs/directives/corres/pdf/i500061_051303/i500061p.pdf>.

[46] Rae A, Robert P and Hausen H-L (1995) *Software Evaluation and Certification: Principles, Practice and Legal Liability*, McGraw-Hill, London, UK.